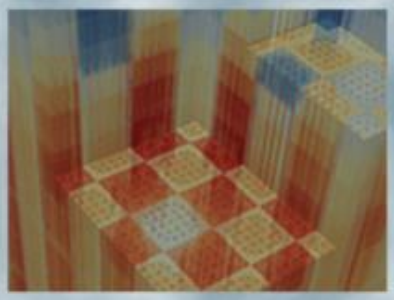
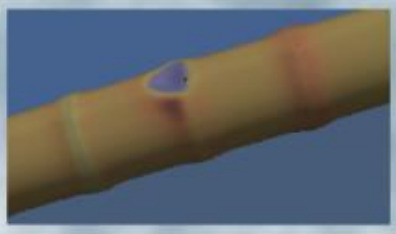
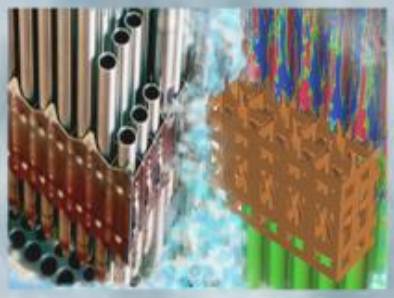
DISCLAIMER

CASL-U-2017-1423-000







MPACT Development Activities at University of Michigan

Brendan Kochunas, University of Michigan
Zack Dodson, University of Michigan
Andrew Fitzgerald, University of Michigan
Mike Jarrett, University of Michigan
Ben Yee, University of Michigan
Shane Stimpson, Oak Ridge National Laboratory

Date: September 29, 2017

REVISION LOG

Revision	Date	Affected Pages	Revision Description
0		All	Initial Release

Document pages that are:

Export Controlled _____

IP/Proprietary/NDA Controlled_____

Sensitive Controlled_____

Requested Distribution:

To:

Copy:

EXECUTIVE SUMMARY

This report documents the efforts of general development tasks in MPACT performed primarily at the University of Michigan during FY17. These tasks included:

1. Unstructured CMFD

   The unstructured CMFD supports the RIA and BWR challenge problems. It allows MPACT to simulate mixed lattice cores (e.g. 8x8 next to 9x9). This design feature is common to operating BWRs and also the SPERT experiments being used to validate VERA-CS for RIA.

2. Higher order spatial sources for MOC

   The higher order sources are for improving the overall performance of MPACT. Higher order spatial sources in MOC provide accurate solutions with potentially significantly fewer mesh regions, thus speeding up every calculation. This method is also particularly important for BWRs.

3. Improving fine source region visualization

   The improvement of the visualization of the flat source region mesh in MPACT is to support users and communication of models and results from MPACT. VERAView provides visualization of pin-wise information, however MPACT resolves the spatial solution to intra-pin level. The capability provided here will let users visualize the full model geometry and components of the solution at the highest spatial resolution available from MPACT.

4. Automated spatial decomposition

   The automated spatial decomposition removes the burden from the user to know how to spatially decompose the model for MPACT; a task that has been rather tricky in the past. Instead the user may give some number of processors that is a multiple of the axial mesh and MPACT will determine a nearly optimal decomposition that gives better load balance than most decompositions used currently.

5. Optimal relaxation factors for feedback

   The development of optimal relaxation factor is to improve the stability and robustness of problems with feedback. Presently the default feedback values are 0.5, which may or may not be optimal. Without relaxation the fixed point iteration that is currently used is known to be unstable for large problems. The optimal relaxation factors are derived by utilizing a previous Fourier analysis.

From the original milestone there was a scope change where the work on the Intel MIC was deferred for work on the automated spatial decomposition.

CONTENTS

FIGURES

TABLES

ACRONYMS

AMA	Advanced Modeling Applications
BWR	boiling water reactor
CASL	Consortium for Advanced Simulation of Light Water Reactors
CP	Challenge Problem
CTF	COBRA-TF subchannel thermal-hydraulics code
CZP	Cold Zero Power
DOE	US Department of Energy
DOE NE	US Department of Energy Office of Nuclear Energy
FA	Focus Area
HPC	high-performance computing
LWR	light water reactor
MOC	method of characteristics
MPACT	Michigan parallel characteristics transport code
OLCF	Oak Ridge Leadership Computing Facility
ORNL	Oak Ridge National Laboratory
PHI	Physics Integration
PWR	pressurized water reactor
R&D	research and development
RIA	reactivity insertion accident
RMS	root mean square
RTM	Radiation Transport Methods
T/H	thermal-hydraulics
UM	University of Michigan
V&V	verification and validation
VERA	Virtual Environment for Reactor Applications

1. INTRODUCTION

1.1 Background

This milestone is given a general scope to complete pre-defined tasks that may not be logically placed under other milestones or because they address the earlier stages of long-term efforts. Some activities may not be placed under other milestones for a number of reasons; these include: the appropriate scoping milestone was not approved or does not exist, the activity cross-cuts several efforts, the activity is in support of student research, or is otherwise a task that requires sufficient effort to be named in a milestone.

This results in a milestone scope that is generally not focused, but rather more of a “grab bag” of tasks. The tasks are generally selected to support research, challenge problems and some user needs.

1.2 Objectives

1.2.1 Scope

The original objectives of this milestone is to complete each of the individually identified tasks:

1. Implementation of Unstructured CMFD
2. Implementation of Higher Order Spatial Sources for MOC
3. Improvements to fine source region visualization
4. Port of the MOC kernels to the Intel MIC
5. Optimal Relaxation Factors for problems with Feedback

The motivation for each of these tasks is given in each of the corresponding sections of this report. However, they are also summarized here:

1. Unstructured CMFD supports the RIA and BWR challenge problems. It allows MPACT to simulate mixed lattice cores (e.g. 8x8 next to 9x9). This design feature is common to operating BWRs and also the SPERT experiments being used to validate VERA-CS for RIA.
2. Incorporating higher order sources in MOC aims to improve the overall performance of MPACT. These intend provide accurate solutions with significantly fewer mesh regions, thus speeding up every calculation. This method is also particularly important for BWRs to avoid having a lot of mesh in the inter-assembly gaps and large water rods.
3. Improvements to the fine source region visualization in MPACT provide support to users and are important in communicating models and results from MPACT. VERAView allows visualization of pin-wise information; however, MPACT resolves the spatial solution to sub-pin level. The capability provided here will let users visualize the full model geometry and components of the solution at the highest spatial resolution available from MPACT.
4. The motivation for porting the MOC kernels to the Intel MIC is to prepare MPACT for being able to take advantage of next generation DOE and NSF leadership computing hardware. Several new leadership class machines are being deployed at the DOE National Labs and universities that use the Intel MIC. These machines are: Theta (at ANL), Cori (at LLNL), and Stampede (at TACC).
5. The goal of optimal relaxation factors with respect to feedback operations is to improve the stability and robustness of coupled problems. Presently, the default feedback values are 0.5, which may or may not be optimal. Without relaxation the fixed point iteration that is

currently used is known to be unstable for large problems. By utilizing the Fourier analysis derived in [1], the relaxation factors may be optimized.

1.2.2 Risks

The primary risk to this milestone was availability of developer time, a secondary risk is availability of computing resources.

1.2.3 Scope Change

Throughout the course of FY17, the previously described scope was modified. The effort that would have been put into the Intel MIC task was instead used to develop an automated spatial decomposition algorithm. The reasons for this scope change included: (1) access to Intel MIC resources was limited, (2) improving the usability of the parallelism for spatial decomposition in MPACT was noted to be a high priority for users, and (3) the automated decomposition would improve the parallel efficiency.

1.3 Summary

The remainder of this document has a section devoted to each of the above stated tasks within the scope. The milestone objective is considered complete as each of the tasks has been completed and merged into the development branch of MPACT.

2. UNSTRUCTURED CMFD

The capability to use CMFD acceleration for the neutron transport on an unstructured coarse grid was developed and implemented in MPACT during FY17. This was done to support a wider range of fuel lattice types, including BWRs and other non-regular lattices such as the Special Power Excursion Reactor Test (SPERT) [2], which is an important problem for testing and verifying the transient capabilities of MPACT.

2.1 Motivation

While many reactors have a regular lattice with a constant pin pitch, this is not always the case. For example, some BWRs will use 7x7, 8x8, 9x9, and/or other fuel assembly types in the same reactor, and the SPERT reactor uses 4x4 and 5x5 fuel assemblies. As a result, the fuel pins do not line up perfectly in a structured 2D Cartesian coarse mesh where each cell has no more than one neighbor on any face. Instead, the coarse cells may be misaligned along certain edges, so a cell may have more than one neighbor on a given face.

The CMFD implementation in MPACT was not originally designed to handle this misalignment, so CMFD acceleration could not be applied as is to these problems. Either the mesh would have to be defined differently (sub-pin level coarse mesh), or CMFD acceleration could not be used. Certain calculations, such as a multicycle BWR depletion, where new assembly types are loaded after the first cycle, were not possible. With the unstructured CMFD implemented in this work, full 3D multicycle depletion of a BWR core in MPACT should be feasible with respect to the neutron transport solver. Considering the building interest for more robust BWR capabilities in MPACT, the importance of this work to CASL is evident.

2.2 Implementation and Limitations

The unstructured CMFD implementation does not support a general, non-Cartesian mesh. All coarse cells must still be rectangular. This essentially follows the same methodology as described in [3]. The simplest example of the type of mesh that would use the unstructured CMFD is given in Figure 2.1. Here, the large coarse cell on the right has two neighbors on the west face.

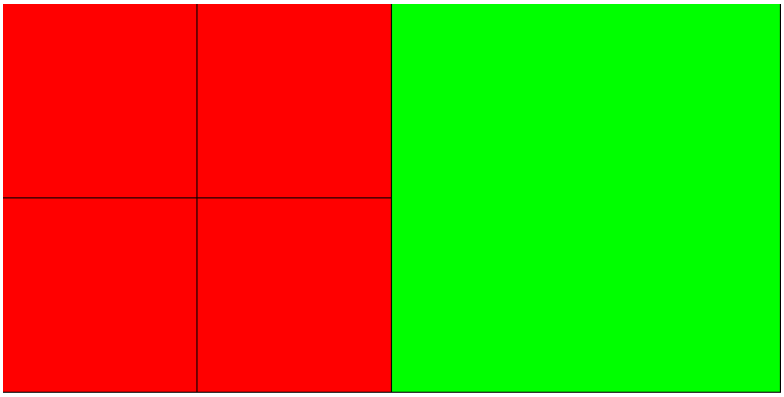


Figure 2.1: Simple Illustration of Unstructured Coarse Mesh

Moreover, there are restrictions on where "unstructured boundaries" like the one shown in Fig. 1 can occur in the CMFD (coarse) mesh. In problems without channel boxes or gaps, unstructured boundaries can be placed at any "module" boundary. However, if the problem has channel boxes or gaps, then these unstructured boundaries can only be placed at lattice boundaries. There is currently no support for unstructured grids inside a module. Furthermore, there is no support for unstructured boundaries in the axial direction. Pins cannot be misaligned in the axial direction.

Finally, the unstructured code only works for product angular quadrature sets for now. The code changes implemented to modify the "prod quad" sweepers would need to be modified to work for "base quad" sets. Since base quadrature sets are rarely used, we are leaving this as potential future work.

The equations for unstructured CMFD are not substantially different from the structured CMFD equations. The only difference is that the calculation of surface quantities such as net and partial current must correctly account for multiple neighbors and fractional overlap between neighboring cells. Most of the work involved modifying the code to be more general when performing ray tracing on the coarse mesh and calculating surface currents during the MOC sweep. Previously, neighbor surface identification was handled in many places by the direction of the neighbor (north, east, west, south, top, and bottom). With an unstructured mesh, this is no longer a unique identifier of a surface, so the coarse mesh surface index corresponding to the neighbor was used instead.

As such, modifications were needed to determine the global index of the surfaces intersected by the MOC rays at module boundaries. To uniquely identify the surface index, some code was added to obtain the next global coarse mesh cell index to be traversed by the MOC ray. Knowledge of the current coarse mesh cell index and the next coarse mesh cell index allows us to uniquely identify the global index of the surface intersected by the ray. For surfaces on parallel processor boundaries, the implementation of several MPI `Isend/Irecv` commands were needed to pass the "next global coarse mesh cell index" to/from the appropriate processors. A new array, `iCMSurfAtEachModBdry`, was added to `CrsMeshLongRayType` in `CoarseRayTrace.f90` to store the surface indices crossed by each long ray at all module boundaries.

Additionally, modifications were made to miscellaneous places in the coarse mesh calculations, such as radial transverse leakage, where 4 radial neighbors were assumed.

2.3 Verification of Robustness and Correctness

With the changes introduced by this implementation, it is possible that we could see longer times for ray tracing, as well as longer times for calculating radial transverse leakage, but the changes are not expected to be significant. A few problems were run to verify that the changes did not significantly affect run time. Three different cases using the center UO2 assembly from the 3D C5G7 benchmarks [4] were run. Additionally, VERA Progression Problem 4 was run on Eos to verify the results for a larger problem.

Table 2.1: CPU hours for master and new Unstructured Branch

Case	Master [CPU-hr]	Unstructured Branch [CPU-hr]
C5G7 Unrodded	0.592	0.587
C5G7 Rodded A	0.649	0.644
C5G7 Rodded B	0.684	0.683
Problem 4	23.3	23.4

2.3.1 Testing

Three regression tests and a unit test were added to test the unstructured mesh capability.

- Unit tests added/modified:
- `testMOCSweeper_UnstructuredCoarseMesh` (new)
- `testCoarseRayTrace` (modified)

Regression tests added:

- `MPACT_exe_testValid_spert_2D_unstructured`
- `MPACT_exe_testValid_spert_3D_unstructured`
- `MPACT_exe_testValid_pb_2x2_unstructured`

The new unit test ensures that the MOC sweepers produce the correct result when an unstructured mesh is used. The regression tests verify that MPACT produces the correct solution when unstructured meshes are used. `spert_2D` and `pb_2x2` are 2D cases, while `spert_3D` is a 2D/1D case.

Additionally, one-time testing was done to verify that the changes introduced do not interfere with other capabilities such as multistate, transient, etc.

2.3.2 Visualization in VERAView

VTK visualization is unchanged with the unstructured coarse mesh. VERAView, however, looks slightly different. Figures 2.2 and 2.3 show the VERAView display of the SPERT core (checkerboard with two 4x4+buffer assemblies, two 5x5 assemblies).

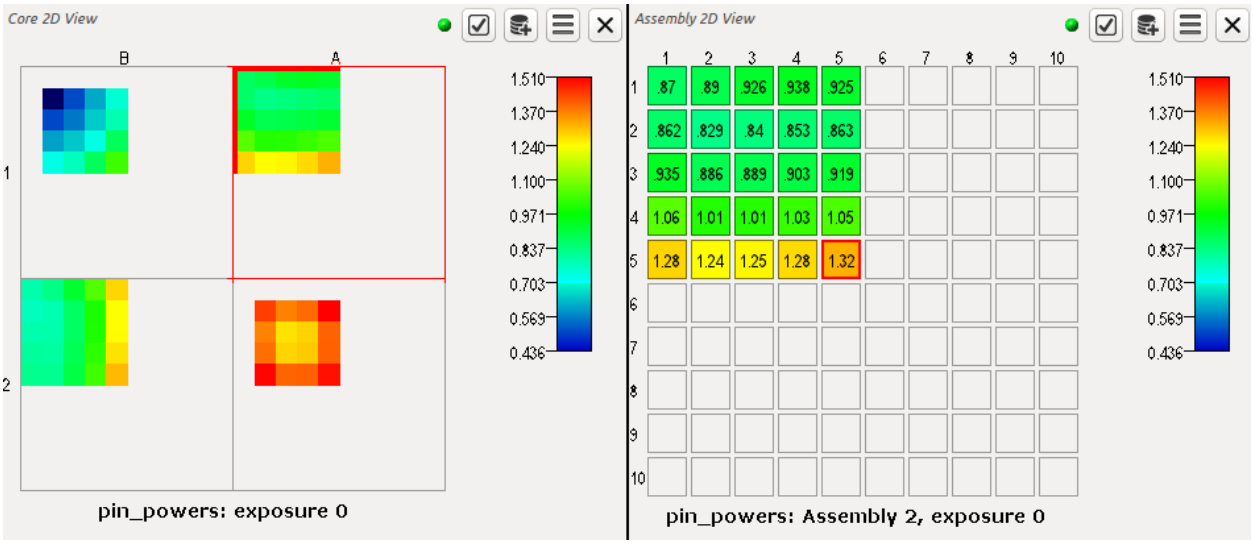


Figure 2.2: VERAView display of SPERT using Structured CMFD, quarter pins

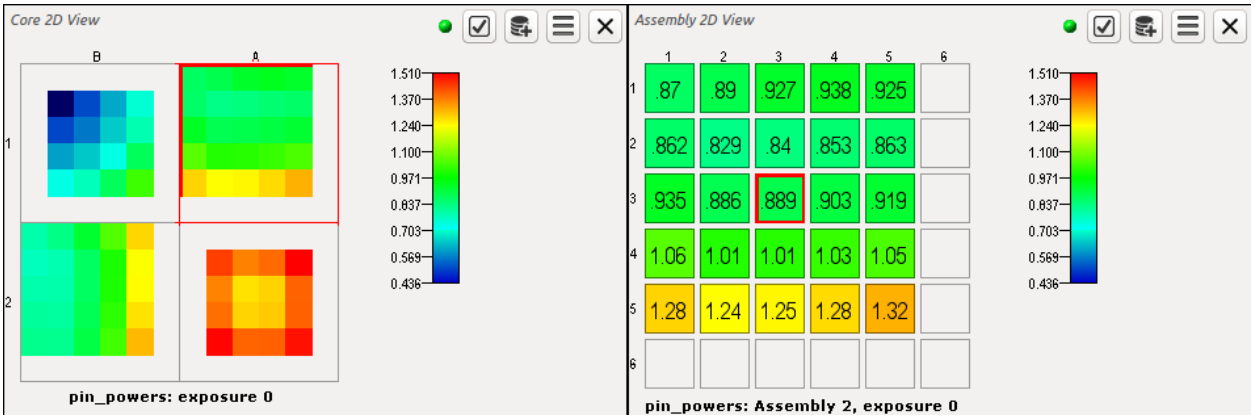


Figure 2.3: VERAView display of SPERT using Unstructured CMFD

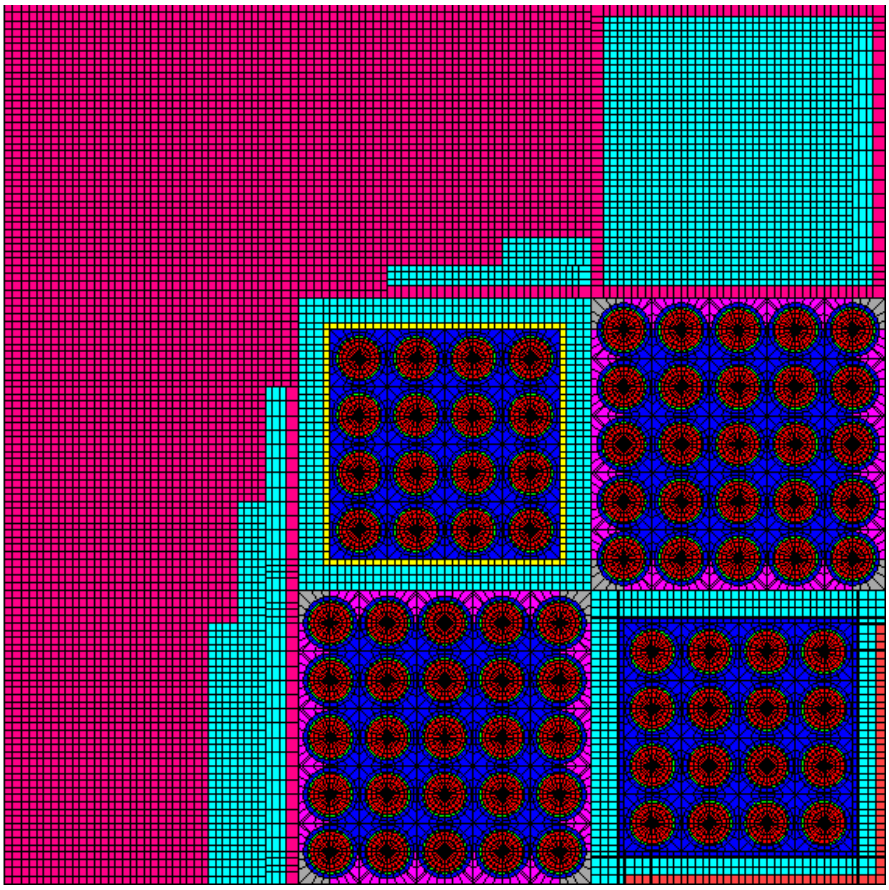


Figure 2.4: SPERT Lattice Geometry

Figure 2.2 displays the pins in a 10x10 lattice because the core is modeled as quarter pins to allow a structured mesh. Because an assembly has 10x10 coarse cells, it is modeled as 10x10 pins in VERAView. In Figure 2.3, the maximum number of "pins" in a lattice is 6x6, which includes the buffer zone around the 4x4 assembly. Thus, VERAView plots both the 4x4 and 5x5 assemblies in a 6x6 lattice. This looks clunky, but it doesn't crash and it does plot the data in a way the user can still interpret it with relative ease. Figure 2.4 shows the VTK edit of the SPERT geometry.

2.4 Future Work

In the future, we may want to improve the visualization in VERAView so that assemblies can be displayed with the correct spatial relationship without false rows of pins when the number of pins per assembly varies. This will require modification of both the HDF5 output and the VERAView script.

3. HIGHER ORDER SPATIAL SOURCES FOR MOC

MPACT's MOC solvers were written using the flat source approximation, however higher-order source-approximations are possible within MOC. MPACT's linear source MOC solvers are based on Ferrer and Rhodes linear source approximation scheme. This scheme calculates linear moments of the angular flux and uses these to compute source moments and source shape with MOC spatial regions. Each MOC segment calculation becomes more expensive when using the linear source approximation, however this approximation allows a model to use a coarser mesh, reducing overall run-time while maintaining accuracy.

3.1 2D Linear Source MOC

3.1.1 Theory

This section outlines the equations used for the 2D linear source approximation MOC solvers implemented in MPACT. These equations are found in Ferrer and Rhodes paper [5], and repeated here.

Segment lengths are renormalized to preserve volume:

$$t_{mki} = s_{mki}\xi_i,$$

with renormalization factor:

$$\xi_i = \frac{1}{\frac{2}{4\pi V_i}\sum_{a/2} \quad \underline{\omega_a}\sum_k \quad t_{mki}},$$

where $t_{mki}$is the normalized segment length of segment k, in region i, in direction m, $s_{mki}$is the unnormalized (geometric) segment length of the same segment. $V_i$is the volume of region i, and $\underline{\omega_a}$is the product of azimuthal quadrature weight of azimuthal angle a and segment width. Similarly, $\underline{\omega_p}$is the product of polar quadrature weight of polar angle $p$and polar sine, and $\underline{\omega_m} = \underline{\omega_a}\,\underline{\omega_p}$.

X and Y directions of rays are given for each polar angle by:

$$a_m^x = cos(\varphi_a)sin(\theta_p),$$
$$a_m^y = sin(\varphi_a)sin(\theta_p).$$

and global X and Y coordinates are given by:

$$X(t_m) = a_m^x t_m/\xi_i \; + \; X_{aki}^{in},$$
$$Y(t_m) = a_m^y t_m/\xi_i \; + \; Y_{aki}^{in}.$$

MOC region numerical centroids can be calculated as:

$$X_c = \frac{1}{4\pi V_i}\sum_{a/2} \quad \underline{\omega_a}\sum_k \quad (X_{aki}^{in} + X_{aki}^{out})t_{aki},$$

$$Y_c = \frac{1}{4\pi V_i}\sum_{a/2} \quad \underline{\omega_a}\sum_k \quad (Y_{aki}^{in} + Y_{aki}^{out})t_{aki},$$

with local coordinates given by:

$$x = X - X_c,$$
$$y = Y - Y_c.$$

The outgoing flux of a segment may be calculated analytically to be:

$$\psi_{mki}^g(t_{mki}) = \psi_{mki}^g(0) + (\frac{q_{mki}^g}{\Sigma_{tr,i}^g} - \psi_{mki}^g(0))F_1(\tau_{mki}^g) \; + \; \frac{\widehat{q_{mi}^g}}{2(\Sigma_{tr,i}^g)^2}F_2(\tau_{mki}^g),$$

where $\psi_{mki}^g(0)$is the incident angular flux for the segment, $\underline{q_{mki}^g}$is the averaged source for the segment, $\widehat{q_{mi}^g}$ is the 1st moment of the source for the angle and region, $\Sigma_{tr,i}^g$is the transport cross-section in the region, and $\tau_{mki}^g$is the optical thickness of the segment. The equations $F_1(\tau_{mki}^g)$, and $F_2(\tau_{mki}^g)$are given by:

$$F_1(\tau_{mki}^g) = 1 - exp(-\tau_{mki}^g),$$
$$F_2(\tau_{mki}^g) = 2(\tau_{mki}^g - F_1(\tau_{mki}^g)) - \tau_{mki}^g F_1(\tau_{mki}^g).$$

These functions are tabulated and interpolated to avoid expensive exponential operations in the internal loops.

During the transport sweep the scalar flux and moments are accumulated using the following forms:

$$\underline{\phi_i^g} = \frac{q_i^g}{\Sigma_{tr,i}^g} + \frac{1}{\Sigma_{tr,i}^g V_i} \sum_m \underline{\omega_m} \sum_k \Delta_{mki}^g,$$

$$\Phi_{ix}^g = \frac{q_{ix}^g}{\Sigma_{tr,i}^g} C_{ix}^g + \frac{q_{iy}^g}{\Sigma_{tr,i}^g} C_{ixy}^g + \frac{1}{\Sigma_{tr,i}^g V_i} \sum_m \underline{\omega_m} \sum_k [a_m^x s_{aki} \psi_{mki}^g(0) H(\tau_{mki}^g) + x_{aki}^{in} \Delta_{mki}^g] t_{mki},$$

$$\Phi_{iy}^g = \frac{q_{ix}^g}{\Sigma_{tr,i}^g} C_{ixy}^g + \frac{q_{iy}^g}{\Sigma_{tr,i}^g} C_{iy}^g + \frac{1}{\Sigma_{tr,i}^g V_i} \sum_m \underline{\omega_m} \sum_k [a_m^y s_{aki} \psi_{mki}^g(0) H(\tau_{mki}^g) + y_{aki}^{in} \Delta_{mki}^g] t_{mki},$$

where $\Delta_{mki}^g = \psi_{mki}^g(0) - \psi_{mki}^g(t_{mki})$,and coefficients are given by:

$$C_{ix}^g = \frac{1}{\Sigma_{tr,i}^g V_i} \sum_{a/2} \underline{\omega_a} \sum_p \underline{\omega_p} \sum_k (a_m^x s_{aki})^2 G_2(\tau_{mki}^g) + \frac{2}{V_i} \sum_{a/2} \underline{\omega_a} \sum_k (x_{aki}^c)^2 t_{mki},$$

$$C_{iy}^g = \frac{1}{\Sigma_{tr,i}^g V_i} \sum_{a/2} \underline{\omega_a} \sum_p \underline{\omega_p} \sum_k (a_m^y s_{aki})^2 G_2(\tau_{mki}^g) + \frac{2}{V_i} \sum_{a/2} \underline{\omega_a} \sum_k (y_{aki}^c)^2 t_{mki},$$

$$C_{ixy}^g = \frac{1}{\Sigma_{tr,i}^g V_i} \sum_{a/2} \underline{\omega_a} \sum_p \underline{\omega_p} \sum_k a_m^x a_m^y (s_{aki})^2 G_2(\tau_{mki}^g) + \frac{2}{V_i} \sum_{a/2} \underline{\omega_a} \sum_k x_{aki}^c y_{aki}^c t_{mki},$$

where $x_{aki}^c$is the x-coordinate of the segment midpoint and equations are given by:

$$G_1(\tau_{mki}^g) = 1 + \frac{\tau_{mki}^g}{2} - (1 + \frac{1}{\tau_{mki}^g}) F_1(\tau_{mki}^g),$$

$$G_2(\tau_{mki}^g) = \frac{2\tau_{mki}^g}{3} - (1 + \frac{2}{\tau_{mki}^g}) G_1(\tau_{mki}^g),$$

$$H(\tau_{mki}^g) = \frac{\tau_{mki}^g}{2} - G_1(\tau_{mki}^g).$$

Source moments are found using these scalar flux moments:

$$\vec{Q}_i^g = \frac{1}{4\pi} [\sum_{g'} \Sigma_{s,i}^{g'} \vec{\Phi}_i^{g'} + \frac{\chi^g}{k} \sum_{g'} \nu \Sigma_{f,i}^{g'} \vec{\Phi}_i^{g'}],$$

where $\vec{\Phi}_i^{g'} = [\underline{\phi_i^g}, \Phi_{ix}^g, \Phi_{iy}^g]^T$. The source expansion coefficients can then be found by solving the following linear system for each region and energy group.

$$Q_{ix}^g = M_{ixx} q_{ix}^g + M_{ixy} q_{iy}^g,$$
$$Q_{iy}^g = M_{ixy} q_{ix}^g + M_{iyy} q_{iy}^g.$$

Where the expansion coefficients can be used to construct source-shape by:

$$q_i^g(x,y) = \underline{q_i^g} + x q_{ix}^g + y q_{iy}^g.$$

The source average and moment along a segment are given by:

$$\underline{q_{mki}^g} = \underline{q_i^g} + x_{aki}^c q_{ix}^g + y_{aki}^c q_{iy}^g,$$

$$\widehat{q_{mi}^g} = \frac{a_m^x q_{ix}^g + a_m^y q_{iy}^g}{\xi_i}.$$

Matrix elements of the linear system are calculated as:

$$M_{ixx} = \frac{1}{4\pi V_i} \sum_{a/2} \underline{\omega_a} \sum_k t_{mki} [(x_{aki}^{in})^2 + (x_{aki}^{out})^2 + a_m^x s_{aki} (x_{aki}^{in} - x_{aki}^{out}) + \frac{2}{3} (a_m^x s_{aki})^2],$$

$$M_{iyy} = \frac{1}{4\pi V_i}\sum_{a/2}\ \underline{\omega_a}\sum_{k}\ t_{mki}[(y_{aki}^{in})^2 + (y_{aki}^{out})^2 + a_m^y s_{aki}(y_{aki}^{in} - y_{aki}^{out}) + \frac{2}{3}(a_m^y s_{aki})^2],$$

$$M_{ixy} = \frac{1}{4\pi V_i}\sum_{a/2}\ \underline{\omega_a}\sum_{k}\ t_{mki}[x_{aki}^{in}y_{aki}^{in} + x_{aki}^{out}y_{aki}^{out} + \frac{s_{aki}}{2}(a_m^x(y_{aki}^{in} - y_{aki}^{out}) + a_m^y(x_{aki}^{in} - x_{aki}^{out})) + \frac{2}{3}a_m^x a_m^y (s_{aki})^2].$$

3.1.2 Implementation

The linear source implementation implements two additional types into MPACT. The new types are for the source, and the MOC sweeper, and are shown in Figure 3.1 and Figure 3.2 respectively. The implementation is currently not compatible with the 2D/1D method, however this is something that can be easily addressed in future work. The functions $F_1(\tau_{mki}^g)$, $F_2(\tau_{mki}^g)$, $G_1(\tau_{mki}^g)$, $G_2(\tau_{mki}^g)$and $H(\tau_{mki}^g)$ are tabulated and interpolated for the MOC sweep routines.

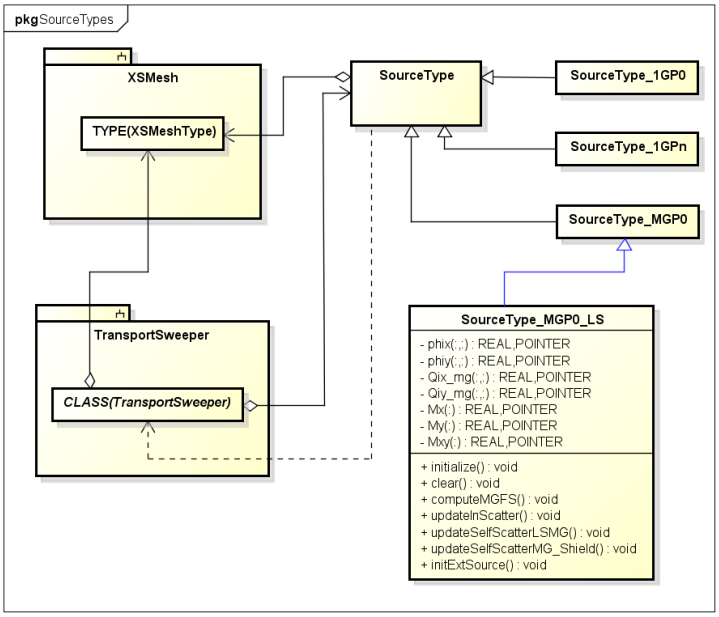


Figure 3.1: MPACT's SourceType structure and use. Newly defined type for linear source: "SourceType_MGP0_LS".

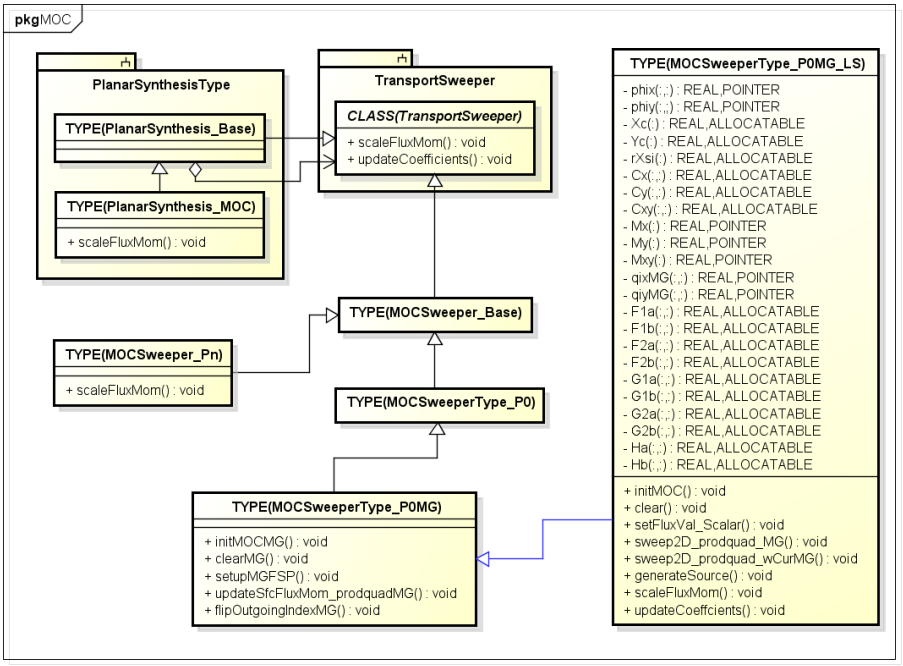


Figure 3.2: Newly added MOCSweeperType for MPACT's linear source based MOC solvers. For the existing types only changes/additions are shown.

3.1.3 Results

The 2D linear source MOC sweeper was tested on several different benchmarks. Results from the 2D C5G7 benchmark are shown in Table 3.1. The coarse mesh cases were run with a single fuel ring with 4 azimuthal divisions, and 8 azimuthal divisions in the coolant. The coarse reflector regions had side-lengths of 0.42 cm. The fine mesh case was run with 5 fuel rings, and 10 coolant rings all with 16 azimuthal divisions. The fine reflector regions had side-lengths of 0.105 cm. Using the linear-source on a coarse mesh was able to achieve results with similar accuracy as using flat-source on the much finer mesh, while reducing total run-time by 20%.

Table 3.1. 2D C5G7 Results comparing flat and linear source approximations.

Scheme	dk (pcm)	Max % Error	RMS Error %	Total Run-Time (s)
FSMOC (Coarse)	64.75	4.2325	1.1581	40.08
LSMOC (Coarse)	1.85	0.6234	0.1617	102.97
FSMOC (Fine)	8.03	0.9681	0.2368	130.65

The 2D B&W 1484 [6] cores results are shown in Table 3.2. Similarly to the C5G7 benchmark, using the linear source gives results similar to that of using the flat source on a much finer mesh.

Table 3.2. Edit 2D B&W Results comparing flat and linear source approximations. The reference solution is using flat-source approximation on a very fine mesh.

Model	FSMOC	LSMOC	Reference
Core 1	1.1641338 (-268 pcm)	1.1668469 (-3 pcm)	1.1668178
Core 2	1.1038155 (237 pcm)	1.1009780 (-47 pcm)	1.1014440

The 2D linear source was also tested on DIMPLE [7],[8], with results shown in Table 3.3.

Table 3.3. 2D DIMPLE Results comparing flat and linear source approximations on coarse and fine meshes.

Experiment	KENO Reference	Source	Source Regions	k-eff (dk pcm)	% RMS	Max. % Abs Diff.
S01A	1.089035	Flat	18350	1.090872 (184)	2.004	5.290
		Linear		1.088684 (-35)	0.348	0.812
		Flat	931455	1.089430 (40)	0.418	0.992
		Linear		1.089257 (22)	0.342	0.792
S06A	1.089035	Flat	18350	1.099348 (292)	2.935	10.680
		Linear		1.096778 (35)	0.419	1.127
		Flat	931455	1.097443 (101)	0.434	1.207
		Linear		1.097411 (98)	0.405	1.127
S06B	1.089035	Flat	18350	1.083907 (34)	0.252	0.659
		Linear		1.084406 (105)	0.266	0.709
		Flat	931455	1.085066 (150)	0.290	0.769
		Linear		1.085074 (151)	0.283	0.749

Future work here involves extending the method to be compatible with MPACT's 2D/1D solvers. Additionally, on the same mesh the linear source approximation is taking about 2.5 times the run-time as flat source. Other groups have seen better ratios than this, so additional effort needs to be spent optimizing this capability.

3.2 3D Linear Source MOC

The 3D linear source in MPACT extends the methods used for the 2D linear source to three dimensions.

3.2.1 Theory

The theory for 3D linear source is essentially the same as the 2D linear source. Some notable differences are in the quadrature weights, where only the angular quadrature weight is used rather than weighting azimuthal and polar angles differently. The linear system to solve for source expansion coefficients may be different than the 2D linear source case. The third dimension gives some flexibility in allowing the source to vary linearly in all directions, only axially, or only radially. These source schemes result in linear systems of size 3x3, 1x1, and 2x2 respectively. Equations remain largely the same, although loops over azimuthal angles will be replaced by loops over angles.

3.2.2 Implementation

The implementation of 3D linear source MOC involved extending the types added during the 2D linear source implementation. Changes to the source type and sweeper type are shown in Figure 3.3 and 3.4 respectively.

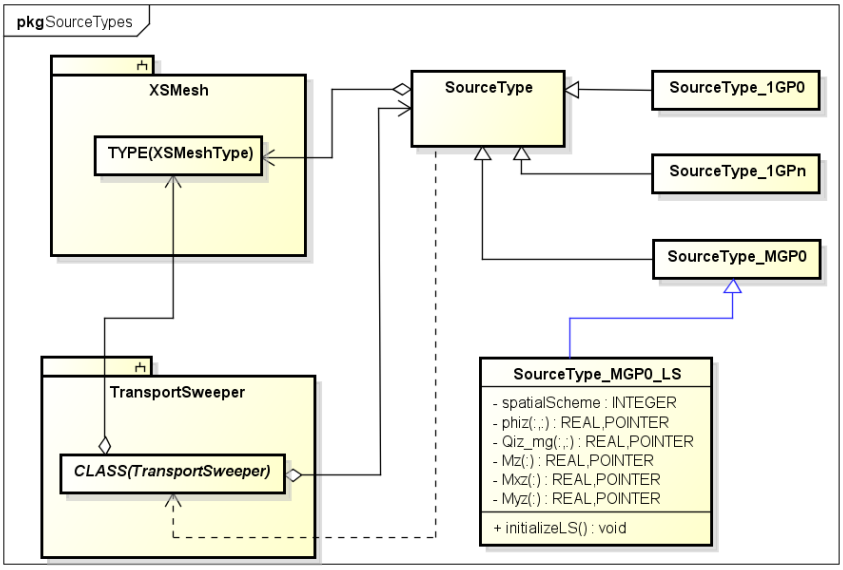


Figure 3.3: Additions/changes to Linear Source type for 3D linear source MOC.

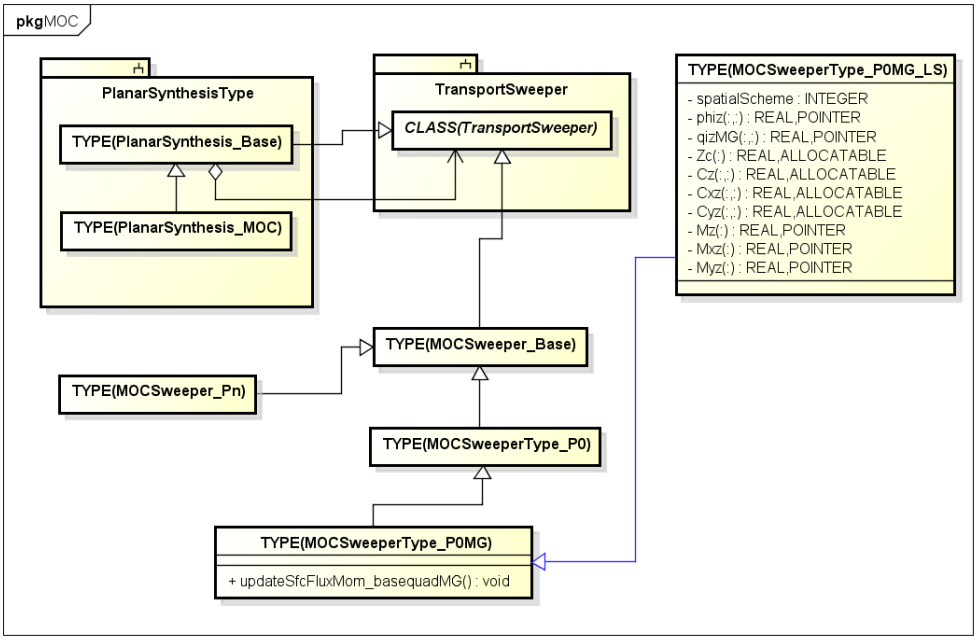


Figure 3.4: Addition/changes to linear source MOC Sweeper type for 3D linear source MOC.

Separate sweep routines were written for each of the different spatial source schemes, with switches on initialization such that only the appropriate moments are allocated. Once the 2D linear source sweepers have been optimized, similar changes should be made to the 3D sweepers as well. After the routines have been optimized further, the different 3D source shape schemes should be tested and compared.

While not yet implemented, theoretical work has been performed to derive a 2nd-order (quadratic) spatial source. The quadratic source will have a shape:

$$q^g_{mki}(s_m) = \overline{q}^g_{mki} + \widehat{q}^g_{mki}\left(s_m - \frac{s_{mki}}{2}\right) + \widetilde{q}^g_{mi}\left(s^2 - s_{mki}s + \frac{s^2_{mki}}{6}\right)$$

The transport equation can be solved analytically, and found to have form:

$$\Psi^g_{mki}(s) = \Psi^g_{mki}(0) + \left(\frac{\overline{q}^g_{mki}}{\Sigma^g_{tr,i}} - \Psi^g_{mki}(0)\right) F_1(\tau) + \frac{\widetilde{q}^g_{mki}}{2(\Sigma^g_{tr,i})^2} F_2(\tau) + \frac{\widetilde{q}^g_{mi}}{(\Sigma^g_{tr,i})^3} F_3(\tau)$$

$$F_3(\tau) = \tau^2 + \tau^g_{mki}\left(\frac{\tau^g_{mki}}{6} F_1(\tau) - \tau\right) - F_2(\tau),$$

where $F_1(t)$ and $F_2(t)$ were previously defined. The 0th angular flux moment will be the same as the flat source case, however the 1st and 2nd order moments must now be considered:

$$\begin{aligned}\widehat{\Psi}^g_{mki} &= \frac{1}{s_{mki}} \int_0^{s_{mki}} s'_m \Psi^g_{mki}(s'_m)\, \mathrm{d}s'_m \\ &= \Psi^g_{mki}(0) s^c_{mki} + \left[\frac{\overline{q}^g_{mki}}{\Sigma^g_{tr,i}} - \Psi^g_{mki}(0)\right] \frac{G_1(\tau^g_{mki})}{\Sigma^g_{tr,i}} + \frac{\widetilde{q}^g_{mki}}{2(\Sigma^g_{tr,i})^2} s_{mki} G_2(\tau^g_{mki}) + \frac{\widetilde{q}^g_{mi}}{(\Sigma^g_{tr,i})^3} s_{mki} G_3(\tau^g_{mki})\end{aligned}$$

$$\begin{aligned}\widetilde{\Psi}^g_{mki} &= \frac{1}{s_{mki}} \int_0^{s_{mki}} {s'_m}^2 \Psi^g_{mki}(s'_m)\, \mathrm{d}s'_m \\ &= \Psi^g_{mki}(0) \frac{s_{mki}}{3} + \left[\frac{\overline{q}^g_{mki}}{\Sigma^g_{tr,i}} - \Psi^g_{mki}(0)\right] \frac{K_1(\tau^g_{mki})}{(\Sigma^g_{tr,i})^2} + \frac{\widetilde{q}^g_{mki}}{2(\Sigma^g_{tr,i})^4} K_3(\tau^g_{mki}) + \frac{\widetilde{q}^g_{mi}}{(\Sigma^g_{tr,i})^5} K_3(\tau^g_{mki})\end{aligned}$$

$$G_3(\tau^g_{mki}) = \frac{\tau^g_{mki} G_1(\tau^g_{mki})}{6} - \frac{1}{4}(\tau^g_{mki})^2 - G_2(\tau^g_{mki})$$

$$K_1(\tau^g_{mki}) = \frac{(\tau^g_{mki})^2}{3} + 2\tau^g_{mki} + 2 - (2\tau^g_{mki} + 2 - \frac{2}{\tau^g_{mki}})F_1(\tau^g_{mki})$$

$$K_2(\tau^g_{mki}) = \frac{(\tau^g_{mki})^3}{2} - (2 + \tau^g_{mki}) K_1(\tau^g_{mki})$$

$$K_3(\tau^g_{mki}) = \frac{-(\tau^g_{mki})^4}{20} + \frac{(\tau^g_{mki})^2}{6} K_1(\tau^g_{mki}) - K_2(\tau^g_{mki})$$

Using these moments one can find the scalar flux and it's moments to be:

$$\overline{\phi}_i^g = \frac{1}{V_i}\sum_m \overline{\omega}_m \sum_k \overline{\Psi}_{mki}^g t_{mki}$$

$$\Phi_{i,x}^g = \frac{1}{V_i}\sum_m \overline{\omega}_m \sum_k \left[x_{mki}^{in}\overline{\Psi}_{mki}^g + \frac{a_m^x \widehat{\Psi}_{mki}^g}{\xi_i}\right] t_{mki}$$

$$\Phi_{i,y}^g = \frac{1}{V_i}\sum_m \overline{\omega}_m \sum_k \left[y_{mki}^{in}\overline{\Psi}_{mki}^g + \frac{a_m^y \widehat{\Psi}_{mki}^g}{\xi_i}\right] t_{mki}$$

$$\Phi_{i,z}^g = \frac{1}{V_i}\sum_m \overline{\omega}_m \sum_k \left[z_{mki}^{in}\overline{\Psi}_{mki}^g + \frac{a_m^z \widehat{\Psi}_{mki}^g}{\xi_i}\right] t_{mki}$$

$$\Phi_{i,x^2}^g = \frac{1}{V_i}\sum_m \overline{\omega}_m \sum_k \left[(x_{mki}^{in})^2\overline{\Psi}_{mki}^g + \frac{2a_m^x x_{mki}^{in}}{\xi_i}\widehat{\Psi}_{mki}^g + \frac{(a_m^x)^2}{\xi_i^2}\widetilde{\Psi}_{mki}^g\right] t_{mki}$$

$$\Phi_{i,y^2}^g = \frac{1}{V_i}\sum_m \overline{\omega}_m \sum_k \left[(y_{mki}^{in})^2\overline{\Psi}_{mki}^g + \frac{2a_m^y y_{mki}^{in}}{\xi_i}\widehat{\Psi}_{mki}^g + \frac{(a_m^y)^2}{\xi_i^2}\widetilde{\Psi}_{mki}^g\right] t_{mki}$$

$$\Phi_{i,z^2}^g = \frac{1}{V_i}\sum_m \overline{\omega}_m \sum_k \left[(z_{mki}^{in})^2\overline{\Psi}_{mki}^g + \frac{2a_m^z z_{mki}^{in}}{\xi_i}\widehat{\Psi}_{mki}^g + \frac{(a_m^z)^2}{\xi_i^2}\widetilde{\Psi}_{mki}^g\right] t_{mki}$$

$$\Phi_{i,xy}^g = \frac{1}{V_i}\sum_m \overline{\omega}_m \sum_k \left[x_{mki}^{in}y_{mki}^{in}\overline{\Psi}_{mki}^g + \left(\frac{a_m^x y_{mki}^{in} + a_m^y x_{mki}^{in}}{\xi_i}\right)\widehat{\Psi}_{mki}^g + \frac{a_m^x a_m^y}{\xi_i^2}\widetilde{\Psi}_{mki}^g\right]$$

$$\Phi_{i,xz}^g = \frac{1}{V_i}\sum_m \overline{\omega}_m \sum_k \left[x_{mki}^{in}z_{mki}^{in}\overline{\Psi}_{mki}^g + \left(\frac{a_m^x z_{mki}^{in} + a_m^z x_{mki}^{in}}{\xi_i}\right)\widehat{\Psi}_{mki}^g + \frac{a_m^x a_m^z}{\xi_i^2}\widetilde{\Psi}_{mki}^g\right]$$

$$\Phi_{i,yz}^g = \frac{1}{V_i}\sum_m \overline{\omega}_m \sum_k \left[y_{mki}^{in}z_{mki}^{in}\overline{\Psi}_{mki}^g + \left(\frac{a_m^y z_{mki}^{in} + a_m^z y_{mki}^{in}}{\xi_i}\right)\widehat{\Psi}_{mki}^g + \frac{a_m^y a_m^z}{\xi_i^2}\widetilde{\Psi}_{mki}^g\right]$$

Using the scalar flux moments one can construct the source, and solve the appropriate linear system (same procedure as for linear source) to find updated source expansion coefficients. Then using a source iteration scheme, one can iterate toward a solution. It is also possible to simplify these scalar flux moments for more efficient execution on a computer.

Currently, the only planned use of the second-order source is in the axial direction, however similar to the linear source it is possible to have various different spatial "schemes". However, any implementation effort on a 2nd-order source solver will not take place until after the linear source has been optimized and brought to production-value.

4. IMPROVING FINE SOURCE REGION VISUALIZATION

4.1 Background

Effective visualization of the Fine Source Regions (FSRs) in MPACT is crucial for observing intra-pin effects. For a single axial level in an MPACT model, each pin contains multiple regions, *O*(100) in which the source is considered to be flat (unless using the recently developed linear source

feature, but for consistency these regions are referred to as FSRs). This represents the finest level of spatial discretization in MPACT. It also leads to a very large amount of output data.

For each FSR, many different quantities can be plotted including energy group-wise flux and reaction rates, and nuclear parameters such as delay group-wise decay constants, etc. When performing FSR edits, the file that MPACT generates is of the Visual ToolKit (VTK) format, which contains several different sub-formats that pertain to different mesh types.

Prior to the completion of this effort, MPACT would output visual edits only to the legacy VTK format, which is a single text file that must be written in serial. This file has the capability to be read by readily available visualization programs such as VisIt or ParaView, although they must read this file in serial as well. These programs allow for manipulation of the 2D or 3D object that is represented and interpretation of the data in each FSR. Geometry in VTK files is discretized as cells bounded by a several vertices. Each of these points is assumed to connected by a straight line. Thus, curved objects such as fuel pins much be approximated with a series of connected straight lines.

For very large MPACT models that necessitate hundreds to thousands of cores, writing the FSR data to a single file was not feasible. Such a file would be difficult to move and is extremely slow to open in a visualization program (if it can be opened at all). Therefore, a solution was required to allow for the visualization of these large models. The most obvious approach was to utilize the existing machinery that is used to parallelize MPACT to assign each core a section of the geometry and the associated FSRs.

4.2 Design and Approach

When executed in parallel, MPACT breaks up the geometry of the model into pieces. Each of these pieces is assigned to a core to perform the relevant transport calculations. More specifically, each core is allocated a portion of the modular geometry mesh. This mesh can be generalized to contain any type of sub-mesh, but each modular mesh partition generally contains all the FSRs for a quarter fuel assembly. Therefore, each core is effectively assigned a spatially contiguous batch of FSRs.

The approach of this task issue was to have each core print the data for each of its FSRs to a separate file. This has the added advantage of dramatically reducing the amount of time to export the VTK data, which was previously in serial. FSR data is accessed through the portion of the modular mesh that is assigned to a given core. This scheme is shown in Figure 4.1. Because the legacy VTK format would not allow for fragmentation of the data in this way, a different format was used. The more modern e**X**tensible **M**arkup **L**anguage (XML)-based format allows for data fragmentation into separate files, each of which can be separately visualized or viewed together. This modern form of VTK has many different sub-formats that correspond to the sub-formats of the legacy VTK format. The sub-format chosen is specific to unstructured grids and is denoted as VTU. In VTU files, the points that compose each cell are listed along with data pertaining to how the points are connected to form cells. Additionally, the value contained in each cell is listed. VTU format is appropriate for MPACT due to the heterogeneous nature of the geometry.

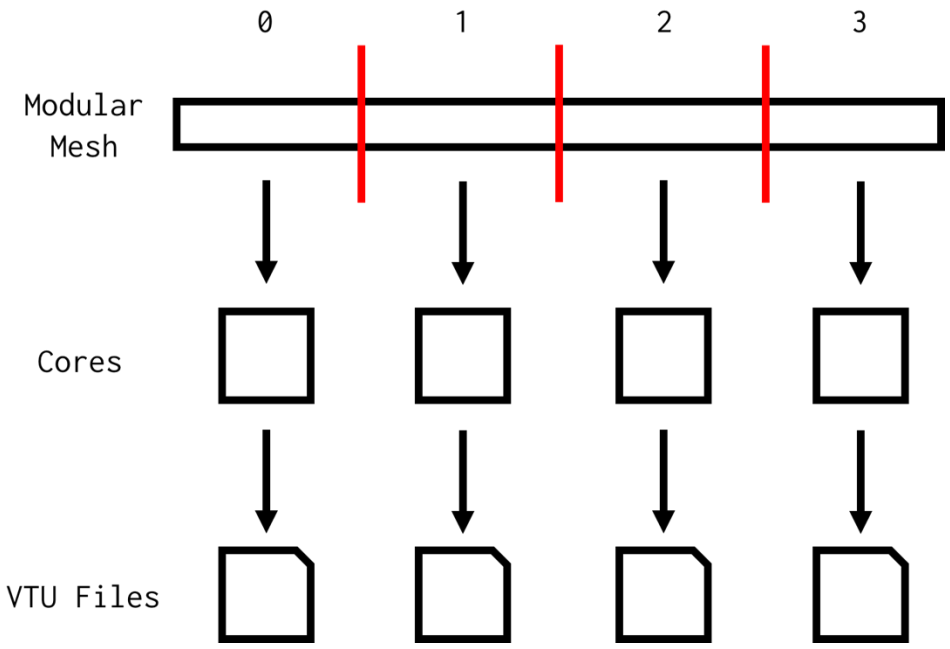


Figure 4.1: VTU File Scheme

MPACT source code was first modified to include the capability to export FSR data using the modern VTU format rather than legacy. This primarily consisted of creating a new file object `VTUXMLFileType` that could properly write FSR data in the new format. This new class was placed in `Futility`, the open source portion of MPACT. A higher-level class called `VisPackageVTUXMLType` was also created, which manages conversion of the FSR data to a format that can be written to VTU files as well as the names of the files.

Since many different files are printed for a single MPACT model, information must be provided as to which files belong together in order to visualize the entire model at once. This is done through an extra file of the type Parallel VTU (PVTU), which contains a list of which files that belong together. Visualization programs can open this file and automatically read from the appropriate files to construct the full output dataset. Another advantage for this file type is the ability to be read in and visualized in parallel. A computer with multiple cores can visualize the data in parallel, which decreases the amount of time to open and manipulate an extremely large dataset (see Peach Bottom in section 4.4 Demonstration).

To test this new functionality, two unit tests and a regression test were added to ensure that the new file format is written properly. The unit tests ensure that the new file object can properly write a sample set of data in the correct format. In the regression test each pin was treated as an “assembly” and each of the different edit types were tested.

The names of tests that were added for this feature are listed below:

- `Futility_testVTUFiles_MPI_1`
- `MPACT_libsUI_testVisPackageVTU_MPI_4`
- `MPACT_exe_Regression_fsr_rr_edits`

4.3 Usage from VERAIn

This feature has full functionality with the VERA input format, but also has limited functionality for the MPACT standard input. To print the model geometry in VERA input files, the `vis_edits` option should be set to `fsrvtu`. For the MPACT standard input, a third boolean should be set to `T` on the `vis_edits` option. For VERA input only, the user can specify the types of data and states for which to perform edits. Edit types are specified in the `[EDITS]` block (or `[STATE]` block), with the following format:

```
edit_group <name> fsr_<edit type>[_<specifier>]
```

The name of the edit should then be placed in the edit option for each state that the edit will be performed. When defining an edit in the `[EDITS]` block, the edit type is prepended by '`fsr_`' and optionally appended by a specifier.

If no specifier is included for an edit type that is discretized (such as flux or lambda) all groups are printed in the edit. For edit types discretized by energy groups, '`_fast`', '`_thermal`', or '`_2g`' can be appended to include fast, thermal, or both energy ranges, respectively. If using `iso-edits`, the specifier is the element symbol followed by A for the isotope; for example: U-235 (for this edit type only, a specifier is required). Table 4.1 shows the available edit types along with their respective available specifiers.

Table 4.1. Edit types and specifiers

Type	Description	Available Specifiers
rr-total	Total reaction rate	energy group index or "2g" if omitted it is energy integrated
rr-absorption	Absorption rate	
rr-fission	Fission neutron production rate	
rr-kappafission	Fission energy production rate	
rr-outscatter	Outscatter rate from the specified group	
flux	scalar flux	
chi	prompt neutron yield	
delayed-chi	delayed neutron yield	
dnpy	macroscopic delayed neutron precursor yield	delayed neutron group
lambda	decay constant	
burnup	fuel burnup	N/A
iso-edits	number density of an isotope	any isotope in model (e.g. U-235)

4.4 Demonstration

Figure 4.2 shows the total reaction rate in the fastest group (1 of 51) from the regression test for this VTU edits; the most thermal group's reaction rate is shown in Figure 4.3. In this example, the cell data for each pin in stored in a separate file.

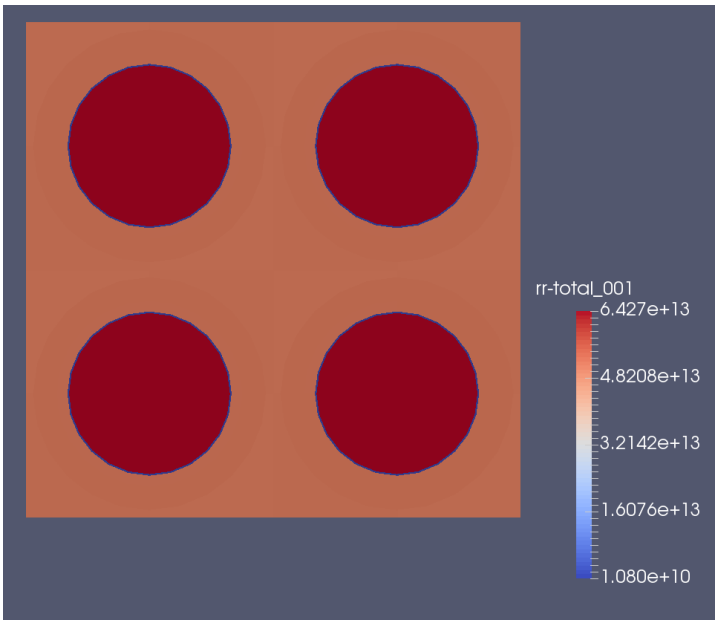


Figure 4.2: Total reaction rate edit of group 1 for regression test problem

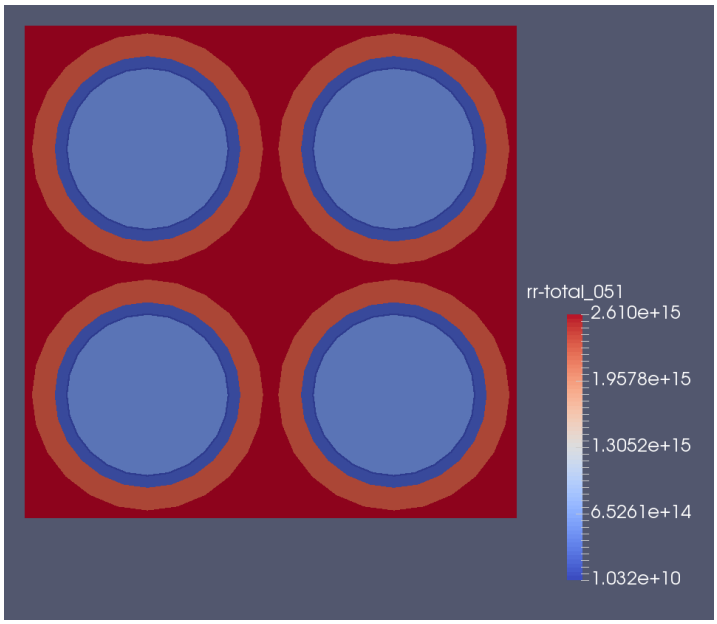


Figure 4.3: Total reaction rate edit of group 51 for regression test problem

Figure 4.4 shows a cutaway of the 3D Peachbottom model generated using the PVTU capability. This model was executed with 3776 cores and so generated 3777 files. It was rendered in parallel on the OCLF RHEA cluster and was rendered in parallel using 32 cores. The time to read the data and render it in VisIt took less than 10 minutes. Each pin is visible in this model, as shown in Figure 4.5.

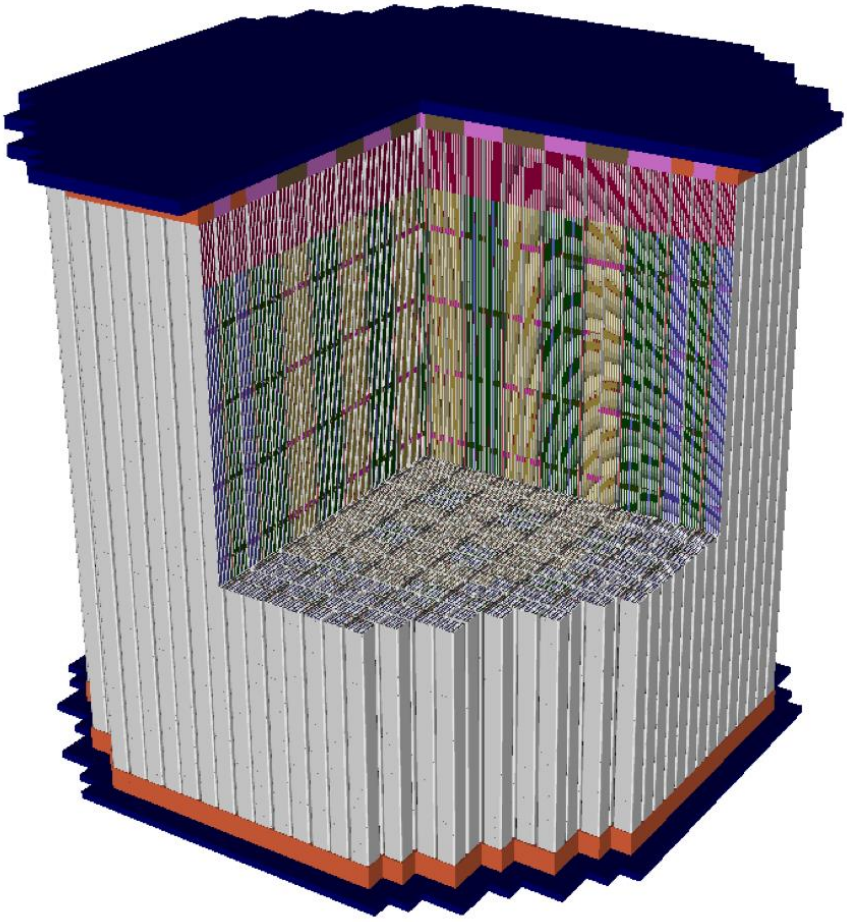


Figure 4.4: Cutaway of the 3D Peach Bottom core executed and rendered in parallel showing materials

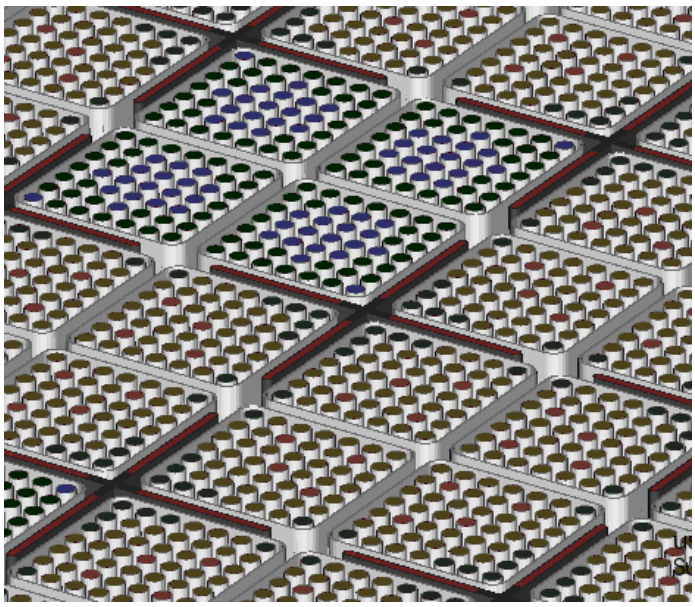


Figure 4.5: A close up of the fuel assemblies of the 3D Peach Bottom model

5. AUTOMATED SPATIAL DECOMPOSITION

5.1 Graph based Decomposition

MPACT is a highly parallel code, and while some forms of decomposition are trivial, spatial decomposition for jagged cores is not. An optimal spatial decomposition will balance the computational work evenly among processes, and minimize the communication between them. Recent changes in MPACT have allowed for more flexible spatial partitions [9], but they must be manually specified. The goal of this work is to automate the process of finding an optimal, or near-optimal spatial decomposition.

The automated spatial decomposition in MPACT relies on abstracting the reactor into a graph, i.e. a set of vertices and edges connecting those vertices; vertices represent a spatial region. In MPACT, this is a module, which contains some amount of computational work. The edges represent communication between the vertices (modules). In MOC, modules will only communicate with directly adjacent modules with a connecting face. The vertices and edges may have associated weights will allow for better load-balancing and minimized communication respectively. In MPACT the vertex weight is computed using the number of source-regions. Once the reactor has been abstracted to a graph, several different methods, such as the ones outlined below, can be applied to decompose the mesh.

5.1.1 Recursive Spectral Bisection

The Recursive Spectral Bisection (RSB) method is an existing method common in computer science for graph decomposition. This method computes the Laplacian matrix of the graph, and computes eigenvectors of it [10]. Using the eigenvector associated with the second-smallest eigenvalue, or Fiedler vector, the graph can be given an ordering. In the event of ties, larger eigenvectors can be used to give consistent ordering. The ordering may then be looped through, and "cut" at a point where the ratio between the two domains is approximately equally weighted. This process can be repeated recursively until the desired number of domains have been decomposed.

5.1.2 Recursive Inertial Bisection

The Recursive Inertial Bisection (RIB) method is an existing method based purely on the coordinates of the vertices [11]. This method takes the following steps:

1. Transform coordinates so origin is at **weighted** centroid
2. Calculate inertial matrix I
3. Solve eigenvectors of I
4. Compute distance from plane defined by largest eigenvector of I
    a. Use planes of smaller eigenvectors for ties

where $I$is defined by:

$$I_{x,y} = \sum_{i=1}^{n} \quad x_i y_i,$$

where x,y are each some spatial coordinate, not necessarily x and y. Similarly to RSB this method can be performed recursively, bisecting the graph at each step.

5.1.3 Recursive Expansion Bisection

The Recursive Expansion Bisection (REB) method is based on existing methods which expand groups, however uses a set of ordered rules which was developed for this work [12]. This method begins by selecting an initial vertex which is as far as possible from the weighted centroid of the graph. The domain is expanded until a desired size is reached using the following ordered rules:

1. Highest connectivity to the current domain
2. Lowest connectivity to the outside domain
3. Largest within domain "sphere of influence"

4. Largest outside domain “sphere of influence”
5. Smallest distance from starting vertex

A “sphere of influence” is a concept developed as part of this work, which includes vertices around a vertex which either directly neighbor, or neighbor two or more neighbors. Examples are shown in Figure 5.1. Similarly to RSB this method can be performed recursively, bisecting the graph at each step.

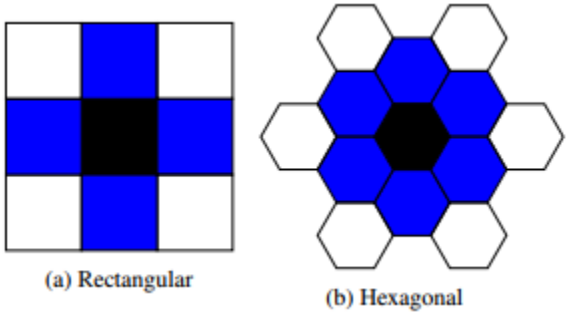


Figure 5.1: Sphere of influence in a graph shown for 2D (a) rectangular (b) hexagonal meshes [12].

5.1.4 Communication Reduction (Refinement)

The methods above are able to get well balanced decompositions, however the communication between them may not be minimal. Methods such as Kernighan-Lin can be used to reduce the communication between bisected domains at each step of the recursive process [13]. Kernighan-Lin (KL) is a method which tests swapping vertices to see if communication can be reduced over several steps, minimizing at each step. This work involved implementing Kernighan-Lin and a modified Kernighan-Lin (SKL) which prioritizes distance between swapped vertices in-order to “straighten” cuts.

5.2 Results

The automated radial decomposition was tested on VERA progression problem 5a-2D. The geometry of this problem is shown in Figure 5.2, with decompositions shown in Figure 5.3.

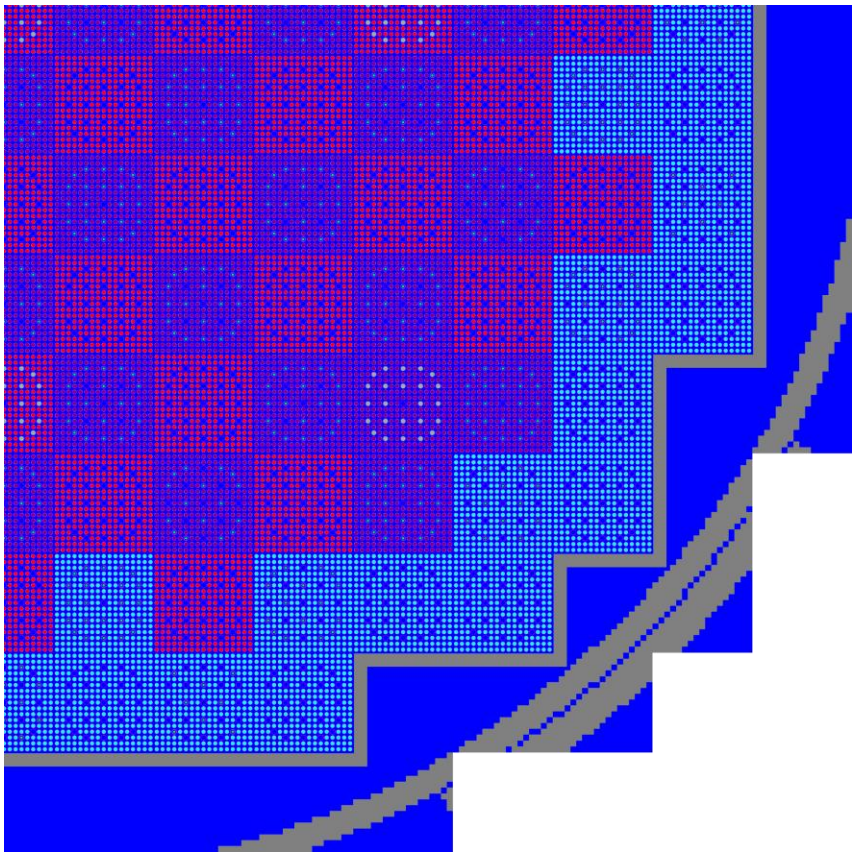
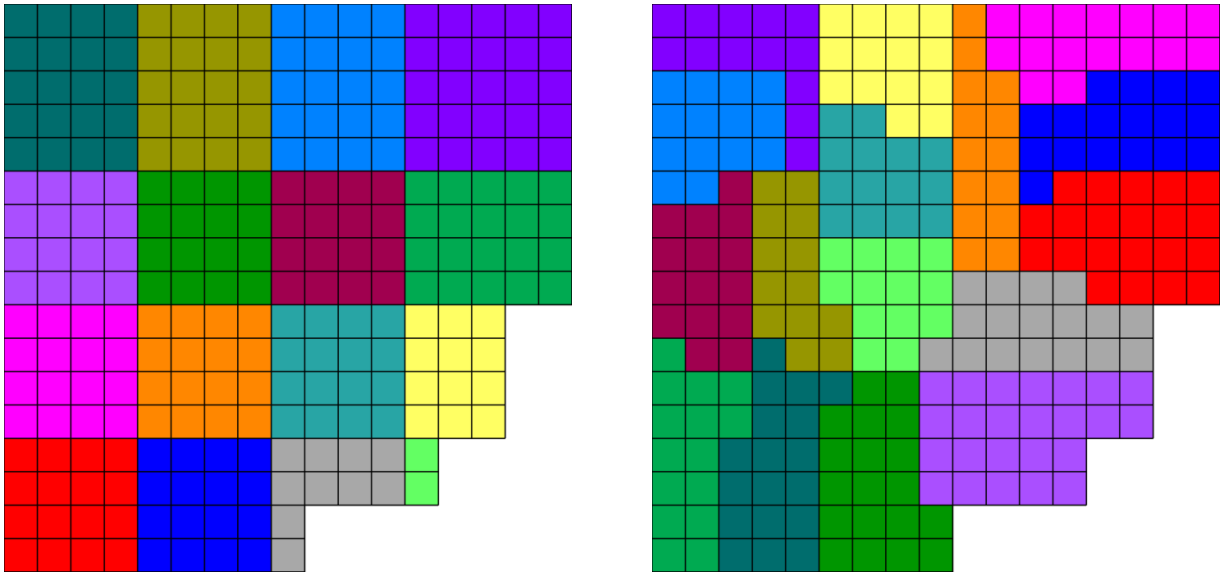


Figure 5.2: Geometry for VERA progression problem 5a-2D.



(a) (b)

Figure 5.2: Spatial decompositions using old (a) Z-order based recursive spectral bisection method, and newly developed (b) REB-SKL method.

Table 5.1. 5a-2D timing results using various decomposition methods.

Method	Time	% Reduction	Min / Max Mem.(MB)
Z-order Recursive Spatial Bisection	916.02	N/A	175.6 / 1481.8
Manually Specified	828.11	9.6	804.1 / 1359.9
RIB	760.94	16.9	1086.4 / 1276.6
REB	697.33	23.9	1078.9 / 1311.2
RSB-SKL	755.67	17.5	1094.5 / 1228.9
RIB-SKL	764.07	16.6	1086.5 / 1276.8
REB-SKL	695.53	24.1	1079.1 / 1311.2

For this progression problem, the REB method with SKL refinement gave the best results reducing run-time from the old Z-Tree methods by 24.1%, and by 16% from the previous best manually specified decomposition. All the methods implemented give near optimal load balancing. However, if refinement methods are not used they may not be convex, which is a current requirement for spatial domains [**Error! Reference source not found.**]. Therefore, it is recommended to always use the refinement method SKL.

Currently, only 2D radial decompositions are available, future work involves extending these methods to generalized 3D domains. With generalized 3D domains, better load-balancing is possible. Especially in cases in which there are axial reflectors, or with staggered-fuel heights.

6. OPTIMAL RELAXATION FACTORS FOR FEEDBACK

Recent work has demonstrated that the current under-relaxation method used in MPACT may become unstable for large problems when feedback is present [1]. A new iteration scheme which is more stable than the current scheme was developed, and is currently being implemented in MPACT. The iteration scheme, written for a 1-group, isotropic scattering, infinite media problem is as follows:

1. Begin iteration $n$with initial guess of scalar flux, $\phi^{(n)}(x)$, net current, $J^{(n)}(x)$, and eigenvalue, $\lambda^{(n)}$.
2. Compute the macroscopic cross sections and diffusion coefficient:
   a. $\Sigma_i^{(n)}(x) = \Sigma_{i0} + \Sigma_{i1}[\phi^{(n)}(x) - \Phi_0]$,
   b. $D^{(n)}(x) = \frac{1}{3\Sigma_t^{(n)}(x)}$.
3. Compute the non-linear current correction factor for low-order diffusion equation:
   a. $\widehat{D}^{(n)}(x) = \frac{d}{dx}[J^{(n)}(x) + D^{(n)}(x)\frac{d\phi^{(n)}}{dx}]/\phi^{(n)}(x)$.
4. Solve the low-order diffusion eigenvalue problem:
   a. $-\frac{d}{dx}D^{(n)}(x)\frac{d\phi^{(n+1/2)}}{dx} + [\Sigma_a^{(n)}(x) + \widehat{D}^{(n)}(x)]\phi^{(n+1/2)}(x) = \lambda^{(n+1/2)}\nu\Sigma_f^{(n)}(x)\phi^{(n+1/2)}(x)$.
5. Perform a transport sweep:
   a. $\mu\frac{\partial\psi^{(n+1)}}{\partial x} + \Sigma_t^{(n)}(x)\psi^{(n+1)}(x,\mu) = \frac{1}{2}[\Sigma_s^{(n)}(x) + \lambda^{(n+1/2)}\nu\Sigma_f^{(n)}(x)][\beta\phi^{(n+1/2)}(x) + (1-\beta)\phi^{(n)}(x)]$.
6. Update the solution for the next iteration:
   a. $\phi^{(n+1)}(x) = \int_{-1}^{1}\psi^{(n+1)}(x,\mu)d\mu\frac{\Phi_0}{\frac{1}{X}\int_0^X\int_{-1}^{1}\psi^{(n+1)}(x',\mu')d\mu'dx'}$,

b. $J^{(n+1)}(x) = \int_{-1}^{1} \mu\psi^{(n+1)}(x,\mu)d\mu' \frac{\Phi_0}{\frac{1}{X}\int_0^X \int_{-1}^{1} \psi^{(n+1)}(x',\mu')d\mu' dx'}$,

c. $\lambda^{(n+1)} = \lambda^{(n+1/2)}$.

Performing the Fourier analysis of this iteration scheme one finds the spectral radius to be:

$$\rho = MAX_{1\leq\omega_j\leq\infty}|\Theta(\omega_j,\beta)|,$$

$$\Theta(\omega_j,\beta) = (1-\gamma)f_{TS}(\omega_j) + \beta[\frac{3}{\omega_j^2}][(1-\gamma)f_{TS}(\omega_j) - 1],$$

$$f_{TS}(\omega_j) = \frac{arctan(\omega_j)}{\omega_j},$$

$$\omega_j = j \cdot \frac{\pi}{\Sigma_{t0}X}.$$(For reflective boundary conditions)

Thus, for each discrete value of $\omega_j$there is a linear in β line, with the spectral radius being the largest magnitude value (at that beta). The optimal value of β can then be found analytically by looking at the intersections between the first discrete value ω_1, and the other values less than a maximum set by NDA.

During feedback calculations, an estimation of the optical thickness of the problem can be estimated during the first iteration, by taking a flux-averaged total cross-section and multiplying by the maximum X,Y,Z dimension, or potentially by using a characteristic 1-D buckling related to the buckling of a right cylinder. The total feedback reactivity coefficient can be found using the difference in eigenvalue from iteration to iteration, and the base nu-fission and absorption cross-sections:

$$\gamma = \frac{d\lambda}{d\phi}\frac{\nu\Sigma_{f0}}{\Sigma_{a0}}.$$

Once this factor has been calculated, the optimal relaxation factor βcan be calculated from a functionalized equation based on the result of the Fourier analysis. A plot of the optimal β is shown in Figure 6.1. There are certain cases, with a sufficiently large or negative γ, which lead to this iteration scheme being unstable. However, these values of γ represent unphysical systems and, therefore are not typical for a reactor. Nevertheless if they do occur MPACT will default to no relaxation ($\beta = 1$).

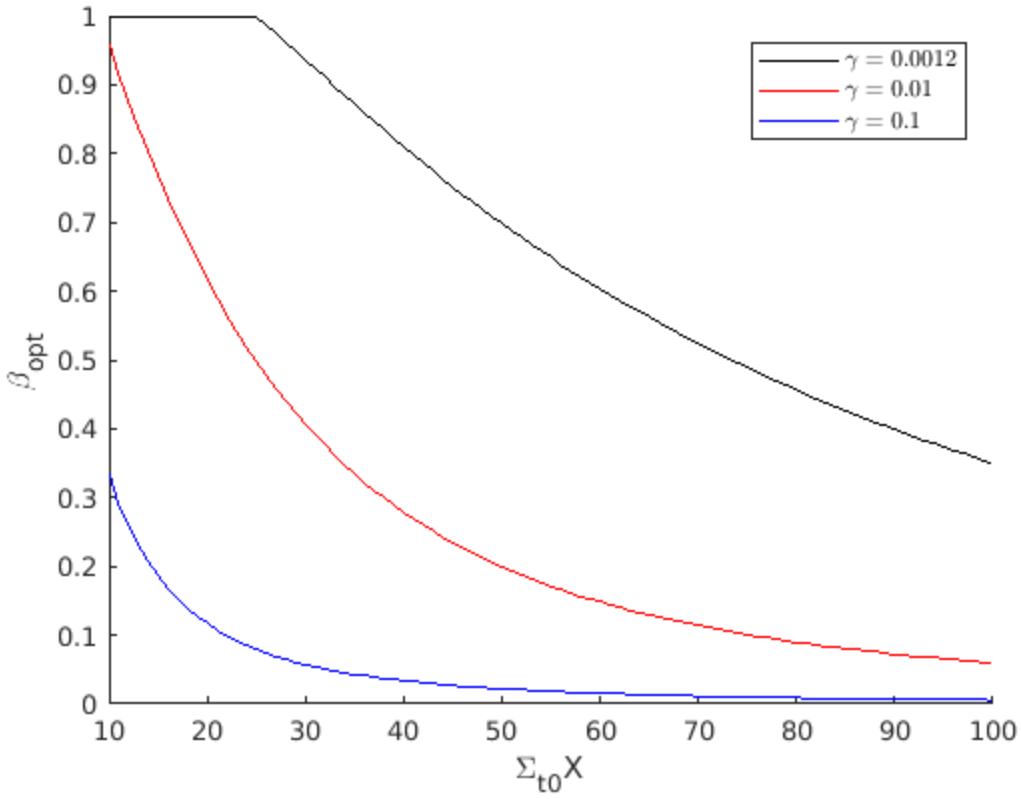


Figure 6.1 Optimal relaxation factor, β, for various values of γ

Future work on this task will involve completing the implementation and testing the effectiveness of this scheme. Then developing a similar result for a more representative iteration scheme; essentially doing the Fourier analysis with a spatial discretization and CMFD.

7. CONCLUSIONS AND FUTURE WORK

The report describes five development activities that were successful completed for MPACT. The completion of these tasks improves the usability, stability and robustness, and capability of the MPACT code for CASL. In a few of the cases, further work is required to realize the full value of these features. By completing the unstructured CMFD capability MPACT is well-suited to tackle the RIA and BWR challenge problems in FY18. The higher order spatial sources have been demonstrated to have a correct implementation, but further work to improve the performance is necessary to realize its full value. The improved visualization edits and automated spatial decomposition were completed and will be providing value in the next release of MPACT. Lastly the optimal relaxation factors demonstrates the potential, however significant future work is still required on this task to have a fully mature capability.

7.2 Future Work

7.2.1 Unstructured CMFD

Future work on the unstructured CMFD will be mostly for maintenance; that is, ensuring this feature continues to work as other features are and code modifications are added. One major effort related to this that still needs to be completed would be better support for edits with unstructured CMFD. However, a lot of this work will need to occur outside of MPACT. Additionally, this feature still requires documentation in the user and theory manuals.

7.2.2 Higher Order Spatial Sources for MOC

At this stage, this effort has achieved a proof of principle. However, the current performance of the implementation is not quite as good as those reported by others. Ongoing work will begin focusing on improving the performance of these kernels, so that models may be run with fewer source regions, and shorter wall times. Additionally, the method needs to be implemented so that is functional for 2D/1D problems. Lastly, as a research effort using higher order spatial sources for the z-direction in 3D MOC calculations should also be investigated. This feature also still requires documentation in the user and theory manuals.

7.2.3 Improving Fine Source Region Visualization

Future work on this task will be mostly for maintenance. There are couple small tasks that are also desirable to complete for usability. Specifically,

1. VisIt apparently has a problem with displaying filled boundary plots for pvtu files. Working with other VisIt users and developers would be needed to resolve this.
2. Operation and usage of how/when the edits are generated when running MPACT. It may be desirable to have a preprocessor or have the edits occur earlier in the execution.
3. Addition of compression to edited files. This is supported in the XML formats, but can also be supported by other file formats such as HDF5 and SILO.

7.2.4 Automated Spatial Decomposition

Future work in this area consists of generalizing the partitioning in full 3D. This will benefit primarily parallel 3D MOC calculations, but will also provide some benefit for 2D/1D calculations. In particular, when axial regions in the model (e.g. the upper and lower reflectors or core plates) have significantly less mesh, a generalized 3D decomposition will achieve a better load balance since the axial planes of the model have a significant difference in the amount of work for each.

7.2.5 Optimal Relaxation Factors for Feedback

The general problem being solved by this effort is improved stability and robustness for coupled simulations, which is something that has been identified as a key area of concern by AMA leadership. The ongoing work here is still in its early stages, and although improved stability has been observed, there is significant value to be gained in pursuing future work on this topic. Specifically the optimal relaxation factors derived thus far do not come from an iteration scheme with a spatial discretization. The next step in this work is to perform the analysis for an iteration scheme more representative of the one used in MPACT (or VERA-CS). This should lead to overall faster convergence rates for the coupled problem. Additionally, we hope to develop a new iteration scheme that is even more rapidly convergent then the ones utilized presently.

REFERENCES

[1] B. Kochunas, A. Fitzgerald, E. Larsen, "Fourier analysis of iteration schemes for k-eigenvalue transport problems with flux-dependent cross sections," *J. Comp. Phys.*, **345**, pp. 294-307, Sept. 2017. https://doi.org/10.1016/j.jcp.2017.05.028

[2] J. Dugone, "SPERT III Reactor Facility: E-CORE Revision," AEC Research and Development, IDO-17036 (1965).

[3] J.W. Thomas, et al., "Assembly Based Modular Ray Tracing and CMFD Acceleration for BWR Cores with Different Fuel Lattices," *Proc. of ICAPP 06*, Reno, NV, USA (2006).

[4] OECD/NEA, "Benchmark on Deterministic Transport Calculations Without Spatial Homogenisation", NEA/NSC/DOC(2003)16.

[5] Rodolfo Ferrer and Joel Rhodes. "A Linear Source Approximation Scheme for the Method of Characteristics," *Nucl. Sci. Eng.*, **182**, pp. 151-165, Feb. 2016. http://dx.doi.org/10.13182/NSE15-6

[6] Critical Experiments Supporting Close Proximity Water Storage Of Power Reactor Fuel, Part I - Absorber Rods, International Handbook of Evaluated Criticality Safety Benchmark Experiments," Tech. Rep. NEA/NSC/DOC/(95)03/IV, OECD Nuclear Energy Agency (NEA) (2014), (B&W 1484 Description in LEU-COMP-THERM-011).

[7] D. Hanlon, et al., "Light Water Moderated and Reflected Low Enriched Uranium (3 wt% ^{235}U) Dioxide Rod Lattices DIMPLE S01," **1**, NEA, NSC, DOC (2006).

[8] D. Hanlon, et al., "Light Water Moderated and Reflected Low Enriched Uranium (3 wt.% ^{235}U) Dioxide Rod Lattices DIMPLE S06," **1**, NEA, NSC, DOC, (2006).

[9] Shane Stimpson and Benjamin Collins. "Flexible Spatial Partitions in MPACT Through Module-Based Data Passing," *Trans. American Nuclear Society* **116** (2017), pp. 654-657.

[10] M. Fiedler. "Laplacian of graphs and algebraic connectivity," *Banach Center Publications*, **25.1**, pp. 57-70 (1989).

[11] V. Taylor and B. Nour-Omid. "A Study of the Factorization Fill-in for a Parallel Implementation of the Finite Element Method," *Intl. J. Numer. Methos. Engrg.*, **37**, pp 3809-3823 (1994).

[12] A.P. Fitzgerald, et al., "Automated Decomposition of a Structured Grid," Trans. Am. Nucl. Sci. (In press), (2017).

[13] B. Kernighan and S. Lin. "An efficient heuristic procedure for partitioning graphs," *The Bell System Technical Journal*, **49.2**, pp. 291-307 (1970).